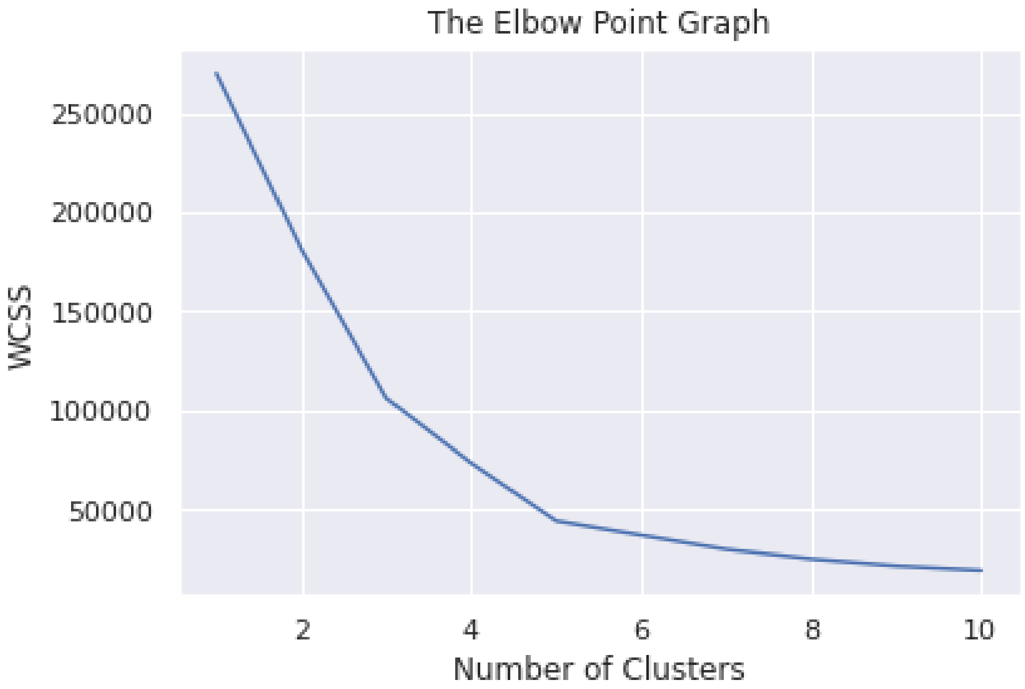
CUSTOMER SEGMENTATION USING DATA SCIENCE

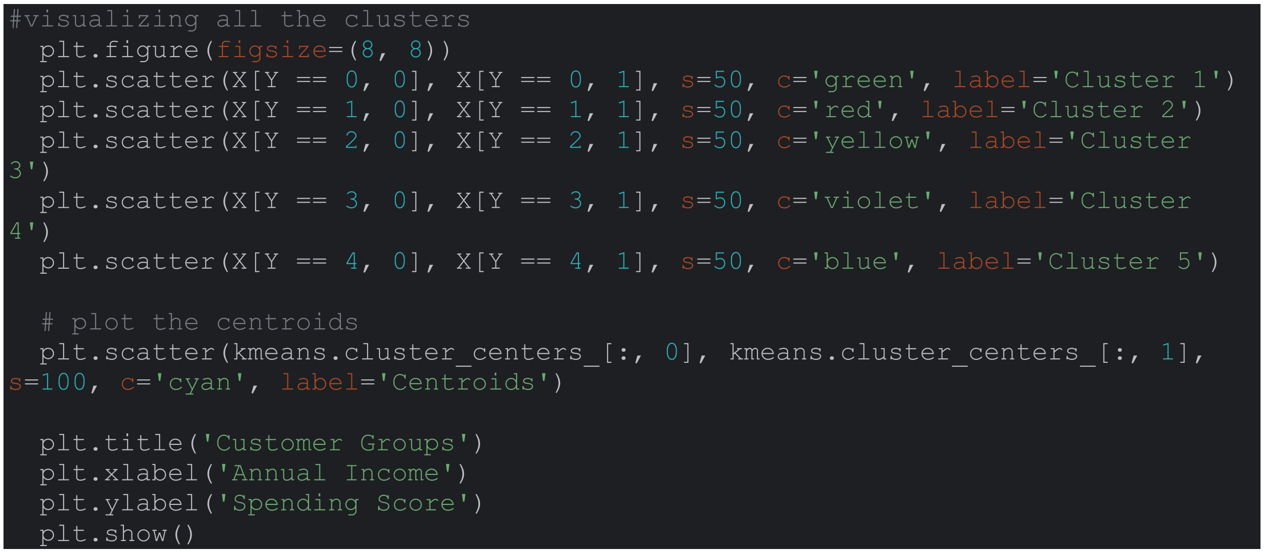
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt


# load the data set
df = pd.read_csv('F:\Mall_Customers.csv')


# Feature Selection
# Select a subset of features for customer segmentation.
selected_features = df[['Age', 'Annual Income (k$)', 'Spending Score (1-
100)']]


# Encoding Categorical Variables
categorical_features = ['Genre']
encoder = OneHotEncoder(drop='first', sparse=False)
encoded_categorical = encoder.fit_transform(df[categorical_features])
encoded_categorical_df = pd.DataFrame(encoded_categorical,
columns=encoder.get_feature_names(categorical_features))


# Concatenate the numerical and encoded categorical features
df_encoded = pd.concat([selected_features, encoded_categorical_df], axis=1)


# Feature Scaling
# Standardize the numerical features.
scaler = StandardScaler()
numerical_features = ['Age', ‘Annual Income (k$)', 'Spending Score (1-100)’]
df_encoded[numerical_features] =
scaler.fit_transform(df_encoded[numerical_features])



# finding wcss value for different number of clusters
#Choosing the Annual Income Column & Spending Score column
X = df.iloc[:,[3,4]].values
wcss = []



for i in range(1,11):
  kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
  kmeans.fit(X)

  wcss.append(kmeans.inertia_)
```

```
# plot an elbow graph

sns.set()
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Point Graph')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```

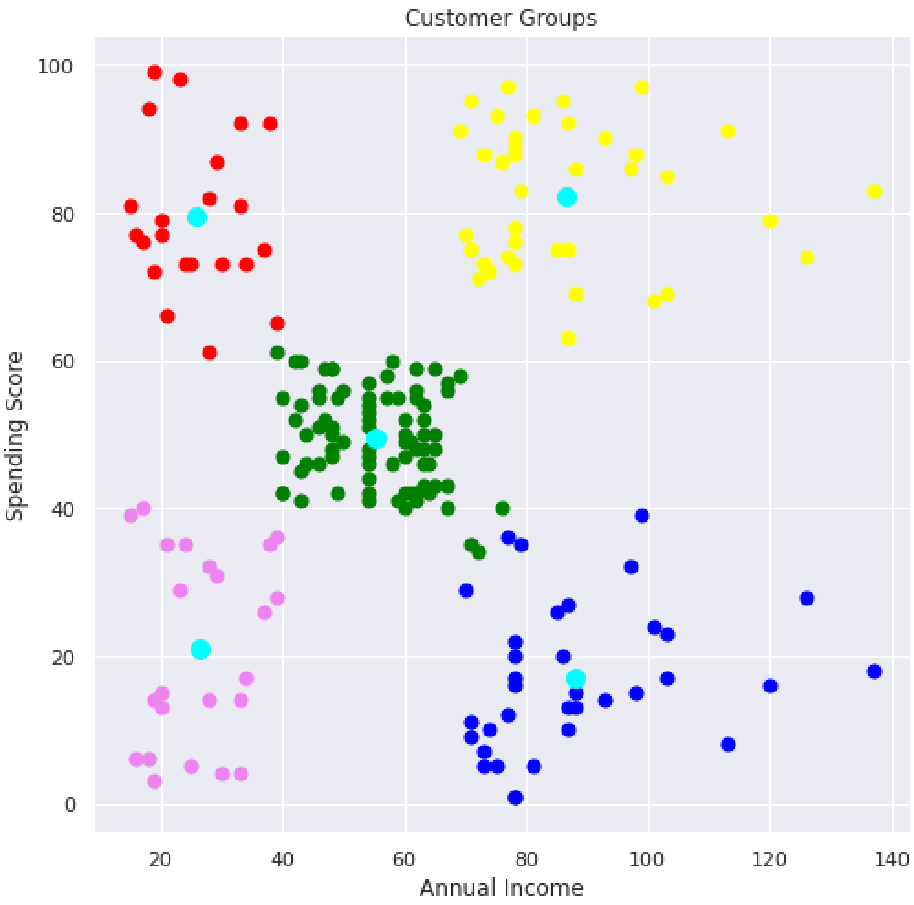


```
#training the k means clustering model
  kmeans = KMeans(n_clusters=5, init='k-means++', random_state=0)

  # return a label for each data point based on their cluster
  Y = kmeans.fit_predict(X)

  print(Y)
```

```
[3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1
 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 0 3 1 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 2 4 2 0 2 4 2 4 2 0 2 4 2 4 2 4 2 4 2 0 2
 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4
 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2]
```



```
#visualizing all the clusters
  plt.figure(figsize=(8, 8))
  plt.scatter(X[Y == 0, 0], X[Y == 0, 1], s=50, c='green', label='Cluster 1')
  plt.scatter(X[Y == 1, 0], X[Y == 1, 1], s=50, c='red', label='Cluster 2')
  plt.scatter(X[Y == 2, 0], X[Y == 2, 1], s=50, c='yellow', label='Cluster
3')
  plt.scatter(X[Y == 3, 0], X[Y == 3, 1], s=50, c='violet', label='Cluster
4')
  plt.scatter(X[Y == 4, 0], X[Y == 4, 1], s=50, c='blue', label='Cluster 5')

  # plot the centroids
  plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s=100, c='cyan', label='Centroids')

  plt.title('Customer Groups')
  plt.xlabel('Annual Income')
  plt.ylabel('Spending Score')
  plt.show()
```



In this project feature engineering involves carefully selecting, creating, and transforming features or variables related to customers, which are essential for distinguishing customer segments effectively. The goal of feature engineering is to extract valuable information and patterns from raw data, making it more suitable for machine learning algorithms

In this k means clustering algorithm were employed this unsupervised machine learning method is employed to group customers with similar attributes or behaviors into distinct segments. The K-Means algorithm divides the customer data into clusters based on feature similarity, with each cluster representing a unique segment. By analyzing customer data, K-Means helps businesses uncover hidden patterns and segment customers effectively